ABSTRACT

An Operating System (OS) function maps affinity to processors for each new task and except for certain circumstances where other processors are permitted to steal tasks, this affinity remains unchanged. Hierarchical load balancing is mapped through an affinity matrix (that can be expressed as a table) which is accessed by executable code available through a dispatcher to the multiplicity of instruction processors (IPs) in a multiprocessor computer system. Since the computer system has multiple layers of cache memories, connected by busses, and crossbars to the main memory, the hierarchy mapping matches the cache memories to assign tasks first to IPs most likely to share the same cache memory residue from related tasks, or at least less likely to incur a large access time cost. Each IP has its own switching queue (SQ) for primary task assignments through which the OS makes the initial affinity assignment. When an IP's SQ becomes task free, the dispatcher code has the free IP look to the SQ of other IPs in accord with the mapped hierarchy, if a threshold of idleness is reached.